SPECIFICATION

Please replace paragraphs [0011], [0012], [0014], [0020] and [0021] with the following amended paragraphs:

[0011] The WPA 100 may be configured with a variety of object-oriented programming languages, such as [[Java]] JAVA® by Sun Microsystems, Inc., Santa Clara, California. An object is generally an item that can be individually selected and manipulated. In object-oriented programming, an object may comprise a self-contained entity having data and procedures to manipulate the data. For example, a [[Java]] JAVA®-based system may utilize a variety of ~~JavaBeans~~ JAVABEANS®, servlets, ~~Java Server Pages ("JSPs")~~ JAVA SERVER PAGES® ("JSPs®"), and so forth. ~~JavaBeans~~ JAVABEANS® are independent, reusable software modules. In general, ~~JavaBeans~~ JAVABEANS® support introspection (a builder tool can analyze how a ~~JavaBeans~~ JAVABEANS® works), customization (developers can customize the appearance and behavior of a ~~JavaBeans~~ JAVABEANS®), events (~~JavaBeans~~ JAVABEANS® can communicate), properties (developers can customize and program with ~~JavaBeans~~ JAVABEANS®), and persistence (customized ~~JavaBeans~~ JAVABEANS® can be stored and reused). [[JSPs]] JSPs® provide dynamic scripting capabilities that work in tandem with hypertext markup language ("HTML") code, separating the page logic from the static elements. According to certain embodiments, the WPA 100 may be designed according to the ~~Java 2 Platform Enterprise Edition (J2EE)~~ JAVA® PLATFORM ENTERPRISES EDITION (J2EE®), which is a platform-independent, [[Java]] JAVA®-centric environment for developing, building and deploying multi-tiered Web-based enterprise applications online.

[0012] The model 12 comprises a definitional framework representing the application state 20. For example, in a web-based application, the model 12 may comprise a ~~JavaBeans~~

JAVABEANS® object or other suitable means for representing the application state 20. Regardless of the application or type of object, an exemplary model 12 may comprise specific data and expertise or ability (methods) to get and set the data (by the caller). The model 12 generally focuses on the intrinsic nature of the data and expertise, rather than the extrinsic views and extrinsic actions or business logic to manipulate the data. However, depending on the particular application, the model 12 may or may not contain the business logic along with the application state. For example, a large application having an application tier may place the business logic in the application tier rather than the model objects 12 of the web application, while a small application may simply place the business logic in the model objects 12 of the web application.

[0014] The view 16 generally manages the visually perceptible properties and display of data, which may be static or dynamic data derived in whole or in part from one or more model objects 12. As noted above, the presentation logic 22 functions to obtain data from the model 12, format the data for the particular application, and display the formatted data to the client 14. For example, in a web-based application, the view 16 may comprise a ~~Java Server Page (JSP page)~~ JAVA SERVER PAGE® (JSP® page) or an HTML page having presentation logic 22 to obtain, organize, format, and display static and/or dynamic data. Standard or custom action tags (e.g., ~~jsp:useJavaBean~~ JSP®:useJAVABEAN®) may function to retrieve data dynamically from one or more model objects 12 and insert model data within the [[JSP]] JSP® pages. In this manner, the MVC architecture 10 may facilitate multiple different views 16 of the same data and/or different combinations of data stored by one or more model objects 12.

[0020] If the client entered data, then the WPA controller 102 creates and populates the appropriate form object 118 as indicated by arrow 154. The form object 118 may comprise any

suitable data objects type, such as a ~~JavaBeans~~ JAVABEANS®, which functions to store the client entered data transmitted via the request 148. The WPA controller 102 then regains control as indicated by arrow 156.

[0021] If the client did not enter data, or upon creation and population of the appropriate form object 118, then the WPA controller 102 invokes the action class 120 to execute various logic suitable to the request 148 as indicated by arrow 158. For example, the action class 120 may call and execute various business logic or WPA logic 126, as indicated by arrow 160 and discussed in further detail below. The action class 120 then creates or interacts with the model object 124 as indicated by arrow 162. The model object 124 may comprise any suitable data object type, such as a ~~JavaBeans~~ JAVABEANS®, which functions to maintain the application state of certain data. One example of the model object 124 is a shopping cart ~~JavaBeans~~ JAVABEANS®, which stores various user data and e-commerce items selected by the client. However, a wide variety of model objects 124 are within the scope of the WPA 100. After executing the desired logic, the action class 120 forwards control back to the WPA controller 102 as indicated by arrow 164, which may be referred to as an "action forward." This action forward 164 generally involves transmitting the path or location of the server-side page, e.g., the [[JSP]] JSP®.